Chapter Four: Contents

(Router – 15 October 2001 – LA-UR 01-5712 – Portland Study Reports)

Chapter Four—Router

1. INTRODUCTION

This chapter documents the input parameters and conditions that were set to generate the initial route set called RS-7. It was generated from activity set AS-7. The general documentation may be obtained from the TRANSIMS web site http://transims.tsasa.lanl.gov. A summary of that document is given here.

Three sets of activities were routed separately: population (AS-7), trucks, and itinerants. The itinerants were further divided into four groups by time of day: a.m., mid-day, p.m., and rest of the day.

The activities generated for the population, trucks, and itinerants are described in the previous chapter. For each pair of successive activities for a given traveler which takes place in different locations, a route through the network is computed. This path is close to the shortest path through the network using the mode specified in the activity file. The section below on link delay noise explains why the path may not be the shortest. The possibly multi-modal path is then divided into unimodal legs and written to the plan file. For each trip, the plan file contains the starting and ending locations, the start time and expected duration of the trip, the distance traveled, and mode-specific path information, such as the nodes passed through for auto trips or route numbers for transit trips.

2. TRAVEL MODES

There are a total of nine transportation modes used in AS-7: auto, walk, bike, transit, light rail, school bus, inter-household shared ride, and park and ride (outgoing and return).

Auto mode (wcw) is used for vehicle trips between activity locations. It is routed as two separate pieces. The first is a walk trip from the source activity location to the current location of the vehicle to be used for the trip (the source parking location). The location of the vehicle is remembered, so that successive auto trips using the same vehicle will use the correct parking location. The second part of the trip is an auto trip from the source parking location to the destination parking location, followed by a walk trip from the destination parking location to the destination activity location. The destination parking location is a parking location connected by a process link to the destination activity location.

Walk mode (w) and bike mode (wiw) are used for foot or bike traffic between two activity locations. They are routed identically with the exception of the speed used—one meter per second for walk mode, and five meters per second for bike mode.

Transit mode (wtw) trips can use buses or light rail for travel between activity locations. Travelers are allowed to transfer between transit routes at a particular stop and to walk between transit stops. Travelers may also walk from the source activity location to the destination activity location if that is faster than using transit vehicles. In order to reduce the number of travelers making transfers that save only a few seconds, a time penalty of 120 seconds is imposed for each transit vehicle boarded. Light rail mode (wlw) is similar to transit mode, with the exception that only light rail routes may be used.

Park and ride has two modes—outbound (wcwtw) and return (wtwcw). Outbound is used for trips leaving home, and return for trips returning to home. An outbound trip is routed similarly to auto, with a walk to the location of the vehicle, followed by a trip that included a drive portion to a park-and-ride parking location, followed by a transit trip to the destination activity location. The transit portion is the same as a transit trip with regard to transfers and walking. A return trip is routed as a transit trip to the location of the vehicle, followed by an auto trip to the destination activity location.

There are two unrouted modes—school bus (wkw) and inter-household shared ride passenger (wKw). Trips of these types are collectively known as “magic move” trips because they are not routed through the transportation network. Instead, the duration is determined from the times of the activities before and after the trip. The Traffic Microsimulator simply moves the travelers from one location to the other.

3. LINK DELAYS

The link delays used for the creation of RS-7 were free-speed delays; that is, the speed at which vehicles would be expected to travel without interference from other vehicles on the road. Starting with RS-8, link delays generated by the Traffic Microsimulator were used.

These delays are the average delay experienced by all of the vehicles on a particular link exiting the link in a 15-minute period. The Route Planner uses these data points, along with the free speed on the link at simulation time 0 and 24 hours, and interpolates between them to generate the delay on a particular link at a particular time.

The problem with this approach is that some links may become completely jammed in a short period of time so that no vehicles exit the link and no data is output by the Traffic Microsimulator. The interpolation by the Route Planner then produces link delay times that are artificially low. To remedy this, the microsimulation configuration file was adjusted to create another output file that lists the links for which no vehicles exited the link during a particular 15-minute interval, but for which there were vehicles present on the link. This allows jammed links to be distinguished from empty links. Link delays were then added for these particular links and time intervals that are the equivalent of the vehicles traveling 0.5 meters per second.

4. OTHER PARAMETERS

4.1 Overdo

Overdo is a heuristic that is intended to reduce the number of possible paths explored by the Route Planner, thus reducing its execution time. The normal search is similar to a circle centered on the source activity location. The size of the circle increases until the destination activity location is found within the circle, as shown in Fig. 1. The overdo parameter changes the shape of the circle to an ellipse, with the long axis of the ellipse aligned with a line connecting the source and destination activity locations, as shown in Fig. 2. The value of the overdo parameter controls the width of the ellipse.

This method works well, in general. However, some geometric configurations in the network cause incorrect paths to be produced. Because of this problem, the overdo parameter was not used for the Portland Study.

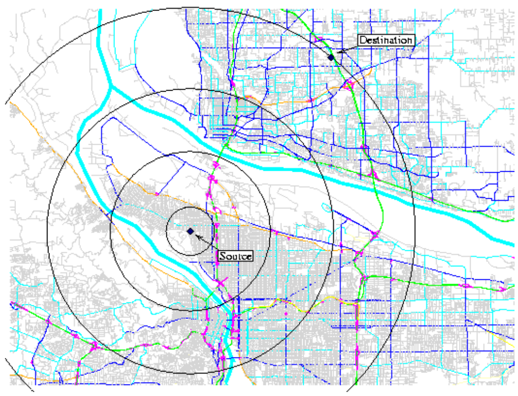

Fig. 1. Search without the overdo parameter.

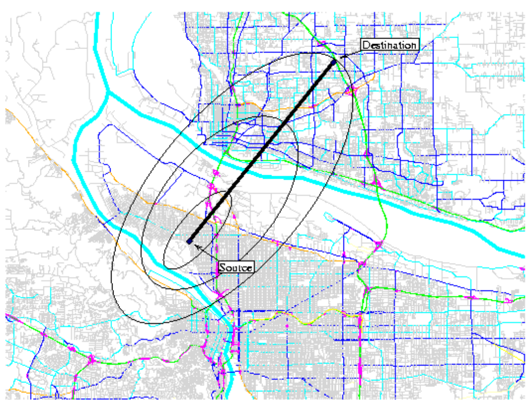

Fig. 2. Search with the overdo parameter.

4.2 Link Delay Noise

Normally, the Route Planner chooses the shortest route between two activity locations. However, there are usually additional paths that are only slightly longer which are viable alternatives. These paths are considered by adding noise to the link delays, so that additional paths are considered.

The link delay noise parameter is specified as a fraction of the link delay to add or subtract. If the delay on a link is d seconds, and the noise factor is f, then the delay used is selected from the uniform distribution $(d-(d*f), d+(d*f))$. A delay noise factor of 0.15 was used to route the population and itinerants, while 0 was used to route the trucks. This ensured that trucks always travel by freeway, which is the shortest path.

4.3 Maximum Leg Time

The maximum leg time parameter is the maximum travel time allowed for a particular leg. While routing, if the current trip exceeds this time, the trip is canceled and an anomaly is reported. For the population, a setting of 10800 seconds (three hours) was used. For trucks and itinerants, a setting of 72000 seconds (20 hours) was used. The large value for trucks and itinerants is used because these trips will never be rerouted during feedback, so it is important to route as many as possible.

4.4 Maximum Trip Time

The maximum trip time parameter limits the start time of a trip. This is done to skip routing of any trips that will not be simulated. A value of 97200 seconds (27 hours) was used for all Portland runs.

5. Execution

The Route Planner is both threaded and distributed. For the Portland Study, AS-7 was routed on 60 nodes. Each node was comprised of two 500 MHz Pentium III processors and 1 Gb of memory. A separate Route Planner process was started on each node. Each process used two routing threads and one I/O thread. The hardware is unstable at times, and 10% to 15% of the nodes would die before the routing was completed—usually during the initialization phase. To deal with this situation, restart functionality was added to the Route Planner.

Each process is independent of the others and has a list of households to route. After each household is routed, its ID is written to the completed household file. If the process crashes after routing has started, the *CreatePartialHHFile* script is run to create a new list of households. This list contains only those households that have not yet been routed. This new household file is used to restart the Route Planner from (approximately) where it left off.

After a router has been restarted, two plan files exist: one from the original router process and one from the restarted router process. If the original router process dies after writing some plans to the plan file, but before completion, the household ID is not written to the completed household file. This may lead to identical plans in both plan files and, in the worst case, an incomplete plan in the original plan file. The *CleanPlan* script removes these duplicate plans and merges the two plan files into one complete plan file.